Diese Kopfleiste bitte unbedingt ausfüllen!

Familienname, Vorname (bitte durch eine Leerspalte trennen)

Bereich	Berufsnummer	IHK-Nummer	Prüflingsnummer
6 6	1 2 0 1		
Sp. 1-2	Sp. 3-6	Sp. 7-9	Sp. 10-14

Termin: Mittwoch, 4. Mai 2022

IHK

Abschlussprüfung Sommer 2022

1201

1 Planen eines Softwareproduktes

Fachinformatiker
Fachinformatikerin
Anwendungsentwicklung (AO 2020)

Teil 2 der Abschlussprüfung

4 Aufgaben
mit Belegsatz
90 Minuten Prüfungszeit
100 Punkte

Hinweis:
Bei der Bearbeitung der Aufgaben ist von einem gewöhnlichen Geschäftsbetrieb auszugehen, der **nicht** durch die COVID-19-Pandemie beeinflusst bzw. durch entsprechende behördliche Verfügungen eingeschränkt ist.

Bearbeitungshinweise

1. Bevor Sie mit der Bearbeitung der Aufgaben beginnen, überprüfen Sie bitte die **Vollständigkeit** dieses Aufgabensatzes. Die Anzahl der zu bearbeitenden Aufgaben ist auf dem Deckblatt links angegeben. Wenden Sie sich bei Unstimmigkeiten sofort an die Aufsicht, weil Reklamationen am Ende der Prüfung nicht anerkannt werden können.
2. Füllen Sie zuerst die **Kopfzeile** aus. Tragen Sie Ihren Familiennamen, Ihren Vornamen und Ihre Prüflings-Nr. in die oben stehenden Felder ein.
3. Lesen Sie bitte den **Text** der Aufgaben ganz durch, bevor Sie mit der Bearbeitung beginnen.
4. Halten Sie sich bei der Bearbeitung der Aufgaben genau an die **Vorgaben der Aufgabenstellung** zum Umfang der Lösung. Wenn z. B. vier Angaben gefordert werden und Sie sechs Angaben anführen, werden nur die ersten vier Angaben bewertet.
5. Tragen Sie die frei zu formulierenden **Antworten dieser offenen Aufgaben** in die dafür lt. Aufgabenstellung vorgesehenen Bereiche (Lösungszeilen, Formulare, Tabellen u. a.) des Arbeitsbogens ein.
6. Sofern nicht ausdrücklich ein Brief oder eine Formulierung in ganzen Sätzen gefordert werden, ist eine **stichwortartige Beantwortung** zulässig.
7. Schreiben Sie deutlich und gut lesbar. Ein nicht eindeutig zuzuordnendes oder **unleserliches Ergebnis** wird als **falsch** gewertet.
8. Zur Lösung der Rechenaufgaben darf ein nicht programmierter, netzunabhängiger **Taschenrechner** ohne Kommunikationsmöglichkeit mit Dritten verwendet werden.
9. Wenn Sie ein **gerundetes Ergebnis** eintragen und damit weiterrechnen müssen, rechnen Sie (auch im Taschenrechner) nur mit diesem gerundeten Ergebnis weiter.
10. Für **Nebenrechnungen/Hilfsaufzeichnungen** können Sie das im Aufgabensatz enthaltene Konzeptpapier verwenden. Dieses muss vor der Bearbeitung der Aufgaben herausgetrennt werden. Bewertet werden jedoch nur Ihre Eintragungen im Aufgabensatz.

Wird vom Korrektor ausgefüllt!

Bewertung

Für die Bewertung gilt die Vorgabe der Punkte in den Lösungshinweisen.

1. Aufg.		Punkte	2. Aufg.		Punkte	3. Aufg.		Punkte	4. Aufg.		Punkte
	15 16			17 18			19 20			21 22	

Prüfungszeit [ ] (23)

Die entsprechende Ziffer (1, 2 oder 3) finden Sie in der Abfrage nach der Prüfungszeit im Anschluss an die letzte Aufgabe.

Gesamtpunktzahl [ ] [ ] [ ] (24 25 26)

Prüfungsort, Datum

Unterschrift

Gemeinsame Prüfungsaufgaben der Industrie- und Handelskammern. Dieser Aufgabensatz wurde von einem überregionalen Ausschuss, der entsprechend § 40 Berufsbildungsgesetz zusammengesetzt ist, beschlossen.
Die Vervielfältigung, Verbreitung und öffentliche Wiedergabe der Prüfungsaufgaben und Lösungen ist nicht gestattet. Zuwiderhandlungen werden zivil- und strafrechtlich (§§ 97 ff., 106 ff. UrhG) verfolgt. – © ZPA Nord-West 2022 – Alle Rechte vorbehalten!

Die Aufgaben 1 bis 4 beziehen sich auf die folgende Ausgangssituation:

Der Energieversorger Wind und Sonne AG möchte den Prozess zur Strom-Abrechnung weiter digitalisieren. Innerhalb dieses Projekts sollen mehrere Apps entwickelt werden. Dabei sollen Sie bei der Planung, Umsetzung und Einführung mitarbeiten und unterstützen.

Korrekturrand

1. Aufgabe (20 Punkte)

Zur Projektdurchführung wählen Sie ein Vorgehensmodell aus. Diese lassen sich in die Kategorien klassisch und agil einordnen.

a) Erläutern Sie, was Sie unter den beiden Kategorien verstehen und nennen Sie jeweils ein Beispiel. 8 Punkte

b) Beschreiben Sie aus der Sicht des Vorstandes des Energieversorgers sowie der mitarbeitenden Stromableser jeweils zwei mögliche Befürchtungen bei der Einführung der Digitalisierungsmaßnahmen und ordnen Sie jeder Befürchtung eine geeignete Gegenmaßnahme zu. Typische Aufgaben für die Stromableser sind die Erfassung der Informationen, die Störungsaufnahme sowie die Kundenberatung. 12 Punkte

Betroffener	Befürchtungen	Maßnahmen
Vorstand des Energieversorgers		
Stromableser		

2. Aufgabe (25 Punkte)

Korrekturrand

Sie schlagen dem Vorstand des Energieversorgers vor, den Stromablesern zur Unterstützung einer effizienten Bearbeitung ihrer Tätigkeiten eine mobile App zur Verfügung zu stellen.

Die benötigte App kann durch die hauseigene IT-Abteilung oder durch einen externen Anbieter entwickelt werden. Dazu wird eine Entscheidungsgrundlage für den Vorstand des Energieversorgers benötigt.

a) Nennen Sie jeweils zwei Vor- und zwei Nachteile einer Fremdvergabe. 4 Punkte

b) Beschreiben Sie drei funktionale und drei nichtfunktionale Anforderungen an die zu entwickelnde App. 12 Punkte

Fortsetzung 2. Aufgabe →

Korrekturrand

c) Für die App müssen die Daten der Stromzähler in einer Datenbank gespeichert werden. Dazu liegt Ihnen der folgende Text zum NoSQL-DBMS MongoDB vor.

Erläutern Sie drei Vorteile beim Einsatz von MongoDB gegenüber relationalen Datenbanken. 9 Punkte

Flexible data models
NoSQL databases typically have very flexible schemas. A flexible schema allows you to easily make changes to your database as requirements change. You can iterate quickly and continuously integrate new application features to provide value to your users faster.

Fast queries
Queries in NoSQL databases can be faster than SQL databases. Why? Data in SQL databases is typically normalized, so queries for a single object or entity require you to join data from multiple tables. As your tables grow in size, the joins can become expensive. However, data in NoSQL databases is typically stored in a way that is optimized for queries. The rule of thumb when you use MongoDB is Data is that is accessed together should be stored together. Queries typically do not require joins, so the queries are very fast.

Easy for developers
Some NoSQL databases like MongoDB map their data structures to those of popular programming languages. This mapping allows developers to store their data in the same way that they use it in their application code.

https://www.mongodb.com/nosql-explained/nosql-vs-sql

3. Aufgabe (30 Punkte)

Korrekturrand

Smart Meter (intelligente Stromzähler) ermöglichen die monatliche Ablesung des Zählerstandes über das Internet. Für die Kunden soll eine Anwendung mit verschiedenen Anzeigen entwickelt werden, welche z. B. die Ablesungen mit Datum und Zählerstand (*ReadingsDisplay*) oder den monatlichen Durchschnittsverbrauch (*AvgDisplay*) anzeigen.

Stromzähler
1HAG1000384497

12.01.2022	22.008 kWh
12.02.2022	22.410 kWh
12.03.2022	22.822 kWh
12.04.2022	23.230 kWh
12.05.2022	23.651 kWh

Stromzähler
1HAG1000384497

Durchschnittlicher monatlicher Verbrauch: **410,75 kWh**

a) Zunächst soll eine Klasse SmartMeter für Stromzählerobjekte modelliert werden.

Die Klasse SmartMeter soll Folgendes beinhalten:
- Die nur klassenintern sichtbaren Instanzvariablen *meterNumber* für die Stromzählernummer und *counter* für den Zählerstand.
- Einen paketübergreifend sichtbaren Konstruktor zur Initialisierung der Instanzvariablen.
- Für den Zählerstand je eine paketübergreifend sichtbare Set- und Get-Methode.

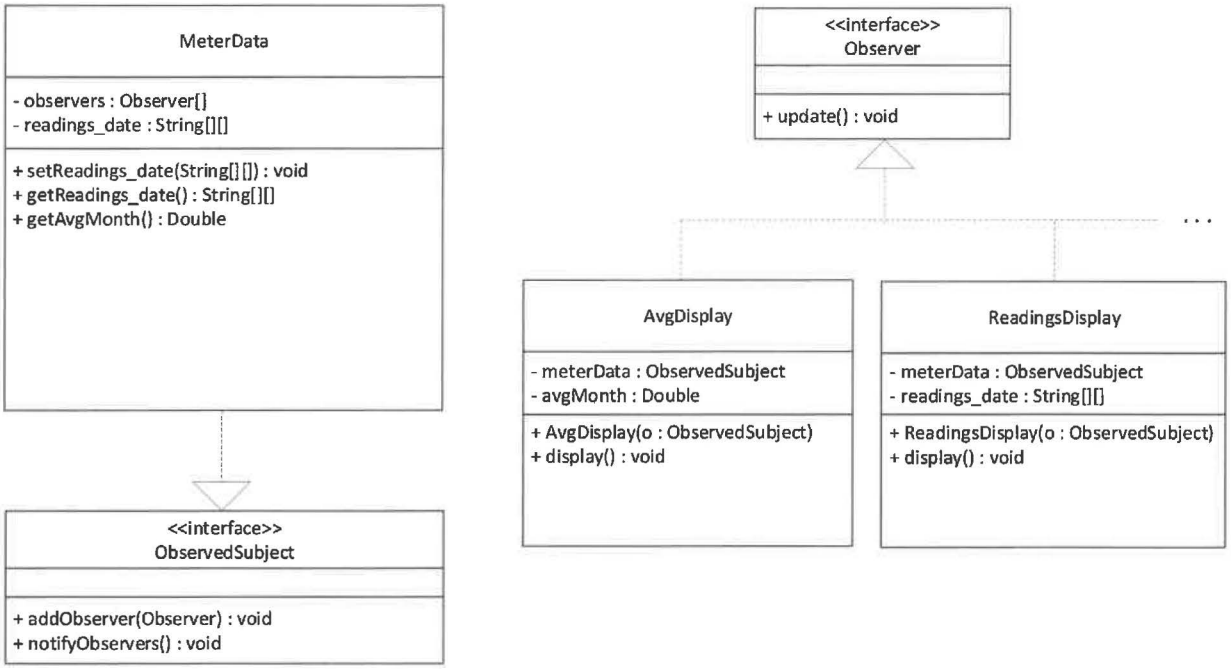
Erstellen Sie das Klassendiagramm für die Klasse SmartMeter.

Hinweis: Geben Sie jeweils auch sinnvolle Datentypen an. 7 Punkte

Fortsetzung 3. Aufgabe →

Korrekturrand

b) Die im jeweiligen Anzeigefenster angezeigten Werte sollen mithilfe des Observer-Musters aktualisiert werden. Ein noch unvollständiger Entwurf zur Umsetzung dieser Anforderung liegt bereits vor:



ba) Ergänzen Sie im Klassendiagramm die Beziehung zwischen der Klasse *MeterData* und dem Interface *Observer* und erläutern Sie diese Art der Beziehung. 4 Punkte

bb) Fügen Sie im Klassendiagramm in den Klassen *MeterData*, *AvgDisplay* und *ReadingsDisplay* die jeweils fehlenden Methodenimplementierungen ein. 2 Punkte

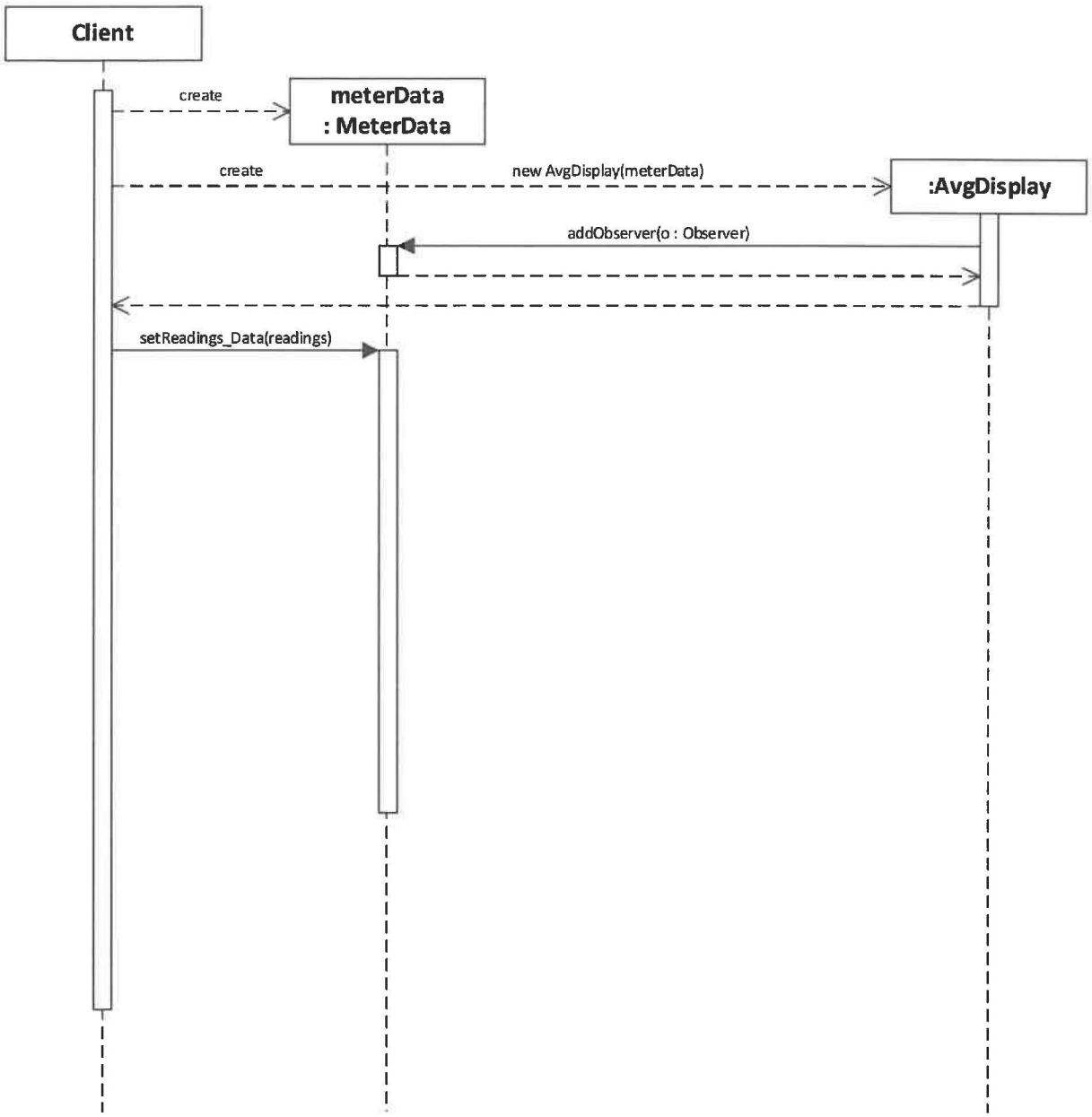
bc) Erläutern Sie die Art der Beziehung zwischen den Klassen *Observer* und *AvgDisplay*. Begründen Sie, warum diese Art der Beziehung hier sinnvoll ist. 4 Punkte

Korrekturrand

c) Der Funktionsweise des verwendeten Observer-Musters wird wie folgt beschrieben:
- Der Client erzeugt ein *MeterData*- und ein *AvgDisplay*-Objekt.
- Im Konstruktor des *AvgDisplay*-Objekts wird die Methode *addObserver* aufgerufen.
- Der Client ruft die Methode *setReadings_Date* auf.
- In der Methode *setReadingsData* wird *notifyObservers* gestartet.
- Die Methode *notifyObservers* führt update aus.
- Die Methode *update* holt sich über den Aufruf der Methode *getAvgMonth* den Double-Wert des monatlichen Stromdurchschnittsverbrauchs und startet die Methode *display* zur Anzeige des Wertes.
- Der Kontrollfluss geht von *display* über *update*, *notifyObservers* und *setReadings_Date* zurück zum Client.

Mithilfe eines Sequenzdiagramms soll der Ablauf der Methodenaufrufe auf den Objekten verdeutlicht werden. Ein unvollständiger Entwurf, in dem die ersten drei Schritte bereits dargestellt sind, liegt vor.

ca) Ergänzen Sie das folgende Sequenzdiagramm gemäß der Ablaufbeschreibung um die noch nicht dargestellten Aufrufe mit allen Methodenbalken, Nachrichten und Antworten inklusive Beschriftung. 9 Punkte



Fortsetzung 3. Aufgabe →

Korrekturrand

cb) Die Methode *notifyObservers* benachrichtigt über den Aufruf von *update* alle registrierten Observer.

Implementieren Sie die Methode *notifyObservers* in Pseudocode. 4 Punkte

4. Aufgabe (25 Punkte)

Korrekturrand

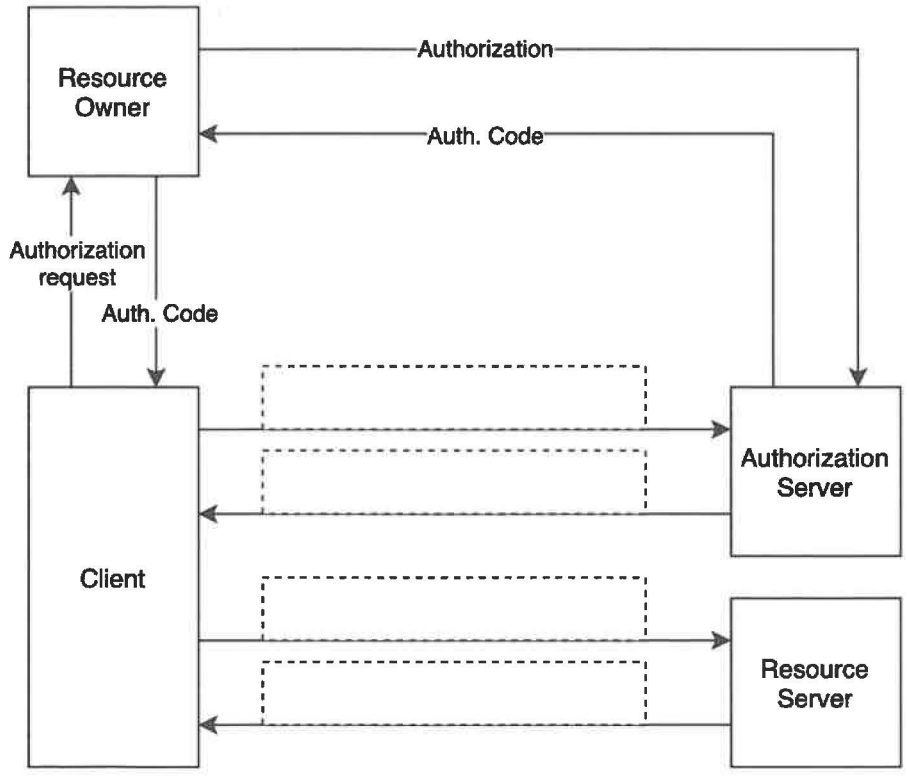
Um für die App einen Support bereitstellen zu können, soll ein Ticketsystem eingeführt werden. Der Energieversorger stellt daher das Ticketsystem auf einem Webserver zur Verfügung.

a) Erläutern Sie drei Vorteile und einen Nachteil beim Einsatz eines Ticketsystems. 12 Punkte

b) Das Ticketsystem verwendet zum Login OAuth2. Dazu liegt Ihnen die nachfolgende Grafik vor, die den Autorisierungs-Prozess beschreibt.

Ordnen Sie die folgenden Begriffe den korrekten Positionen in der Grafik zu, damit der Prozess vollständig beschrieben ist:
- Access Token
- Protected Resource
- Auth. Code

Hinweis: Einer dieser Begriffe muss doppelt verwendet werden. 4 Punkte



Fortsetzung 4. Aufgabe →

Fortsetzung 4. Aufgabe

Korrekturrand

c) Um Datensicherheit und Datenschutz zu gewährleisten, soll der Webserver, auf dem das Ticketsystem installiert werden soll, mit einem Secure Sockets Layer (SSL)-Zertifikat gesichert werden.

Erläutern Sie drei Sicherheitsmechanismen, die durch den Einsatz von SSL-Zertifikaten erreicht werden. 9 Punkte

PRÜFUNGSZEIT – NICHT BESTANDTEIL DER PRÜFUNG!

Wie beurteilen Sie nach der Bearbeitung der Aufgaben die zur Verfügung stehende Prüfungszeit?

[1] Sie hätte kürzer sein können.
[2] Sie war angemessen.
[3] Sie hätte länger sein müssen.

Abschlussprüfung Sommer 2022

IHK

Belegsatz

Fachinformatiker/Fachinformatikerin
Anwendungsentwicklung (AO 2020)
1201

Teil 2 der Abschlussprüfung

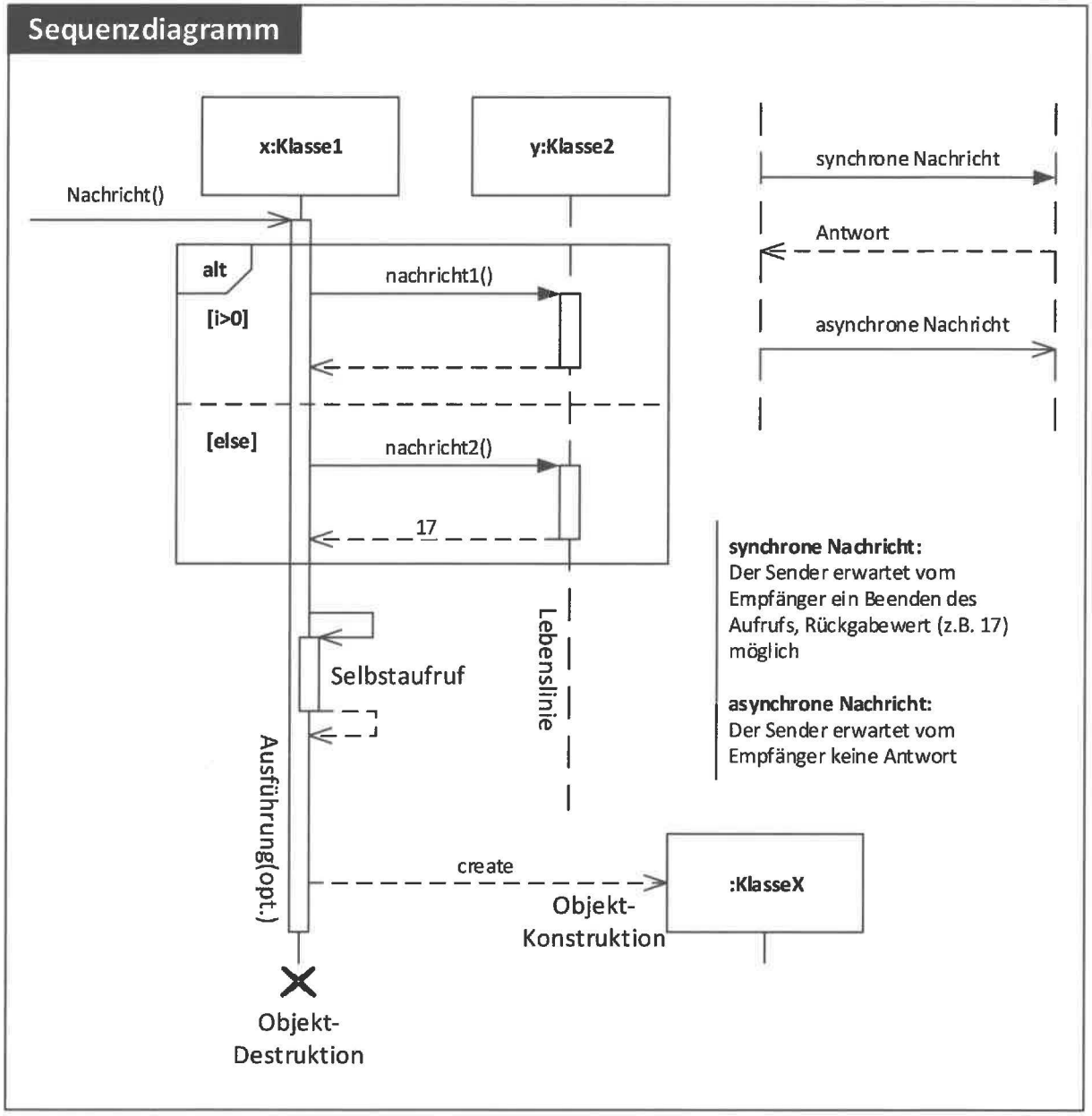
Der Belegsatz ist Grundlage für beide Prüfungsbereiche.

1. Planen eines Softwareproduktes
2. Entwicklung und Umsetzung von Algorithmen

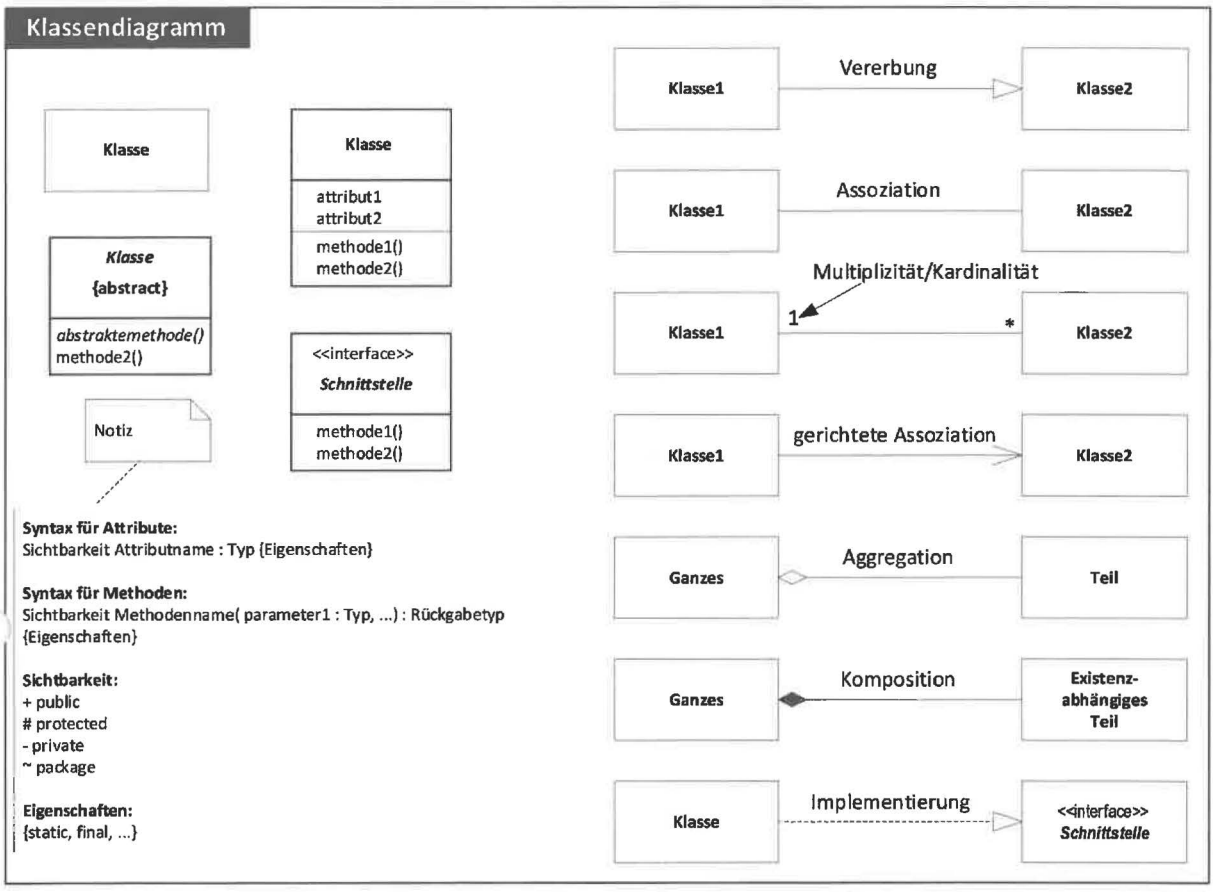
UML-Sequenzdiagramm	2
UML-Klassendiagramm	3
UML-Aktivitätsdiagramm	4
Pseudocode, prozedurale Programmierung	5
Programmablaufplan (PAP) DIN 6601	6
SQL-Syntax (Auszug)	7/8

Gemeinsame Prüfungsaufgaben der Industrie- und Handelskammern. Dieser Aufgabensatz wurde von einem überregionalen Ausschuss, der entsprechend § 40 Berufsbildungsgesetz zusammengesetzt ist, beschlossen.
Die Vervielfältigung, Verbreitung und öffentliche Wiedergabe der Prüfungsaufgaben und Lösungen ist nicht gestattet. Zuwiderhandlungen werden zivil- und strafrechtlich (§§ 97 ff., 106 ff. UrhG) verfolgt. – © ZPA Nord-West 2022 – Alle Rechte vorbehalten!

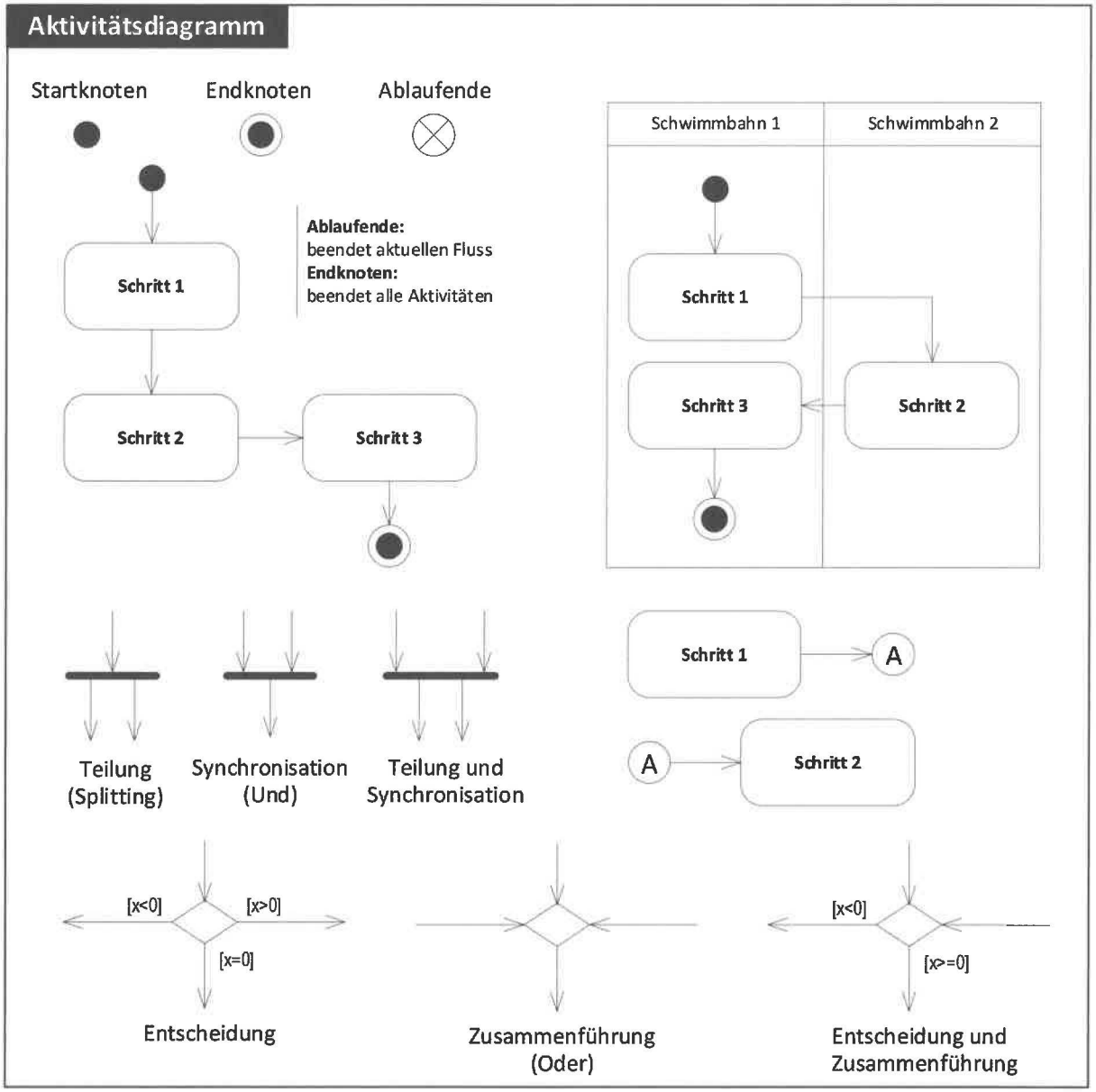
UML-Sequenzdiagramm



UML-Klassendiagramm



UML-Aktivitätsdiagramm



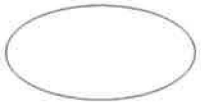
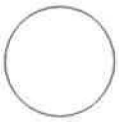
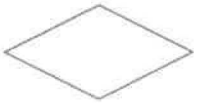
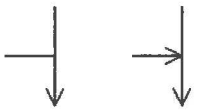
Pseudocode, prozedurale Programmierung

Schriftart: Courier New (12)

Pseudocode	Beschreibung
wenn [Bedingung] **dann** [Anweisung] **sonst** [Anweisung] **Ende wenn**	Verzweigung
zähle [Variable] **von** [Startwert] **bis** [Endwert] [Schleifeninhalt] **Ende zähle**	Zählschleife (Iteration)
solange [Bedingung] [Schleifeninhalt] **Ende solange**	Kopfgesteuerte Schleife
wiederhole [Schleifeninhalt] **solange** < Bedingung >	Fußgesteuerte Schleife
:=	Zuweisung
=; <; >; <=; >=; < >	Vergleiche
integer	
double	
string	
array	

Programmablaufplan (PAP) DIN 6601

Ein PAP eignet sich nicht für den Entwurf von Programmen in Hochsprachen, da Verzweigungen sehr nahe an den Sprungbefehl angelent sind. Für prozedurale Sprachen eignen sich daher eher Struktogramme und für objektorientierte Sprachen UML.

	Grenzstelle Beginn oder Ende eines PAP Text: – „Start" oder „Ende" für das Hautprogramm – Name des Unterprogramms (Start) und „return" (Ende) bei Unterprogrammen.
	Operation Hier können Zuweisungen, einzelne Befehle oder frei sprachliche Beschreibungen der Operationen eingeschrieben werden.
	Unterprogramm Text: Name des aufzurufenden Unterprogramms
	Übergangsstelle Dient der Übersichtlichkeit, da lange Pfeile vermieden werden. Kann nur auf eine Stelle im PAP verweisen, es kann jedoch von mehreren Stellen im PAP auf eine Übergansstelle verwiesen werden. Text: Zusammengehörige Übergangsstellen müssen die gleiche Bezeichnung tragen.
	Verzweigung Text im Symbol: Bedingung, nach welcher verzweigt wird. Text auf den abgehenden Pfeilen: – „Ja" oder „Nein", zwei Pfade – Eine Variable und einen sich verzweigenden Ausgangspfad. An jedem Zweig wird der Wert angegeben, bei dem der jeweilige Weg gewählt wird.
	Ablauflinie
	Zusammenführung ohne und mit Pfeil
	Kommentar

SQL-Syntax (Auszug)

Syntax	Beschreibung
Tabelle	
CREATE TABLE Tabellenname(Spaltenname < DATENTYP >, Primärschlüssel, Fremdschlüssel)	Erzeugt eine neue leere Tabelle mit der beschriebenen Struktur
ALTER TABLE Tabellenname **ADD COLUMN** Spaltenname Datentyp **DROP COLUMN** Spaltenname Datentyp **ADD FOREIGN KEY**(Spaltenname) **REFERENCES** Tabellenname(Primärschlüsselspaltenname)	Änderungen an einer Tabelle: Hinzufügen einer Spalte Entfernen einer Spalte Definiert eine Spalte als Fremdschlüssel
CHARACTER	Textdatentyp
DECIMAL	Numerischer Datentyp (Festkommazahl)
DOUBLE	Numerischer Datentyp (Doppelte Präzision)
INTEGER	Numerischer Datentyp (Ganzzahl)
DATE	Datum (Format DD.MM.YYYY)
PRIMARY KEY (Spaltenname)	Erstellung eines Primärschlüssels
FOREIGN KEY (Spaltenname) **REFERENCES** Tabellenname(Primärschlüsselspaltenname)	Erstellung einer Fremdschlüssel-Beziehung
DROP TABLE Tabellenname	Löscht eine Tabelle
Befehle, Klauseln, Attribute	
SELECT * | Spaltenname1 [, Spaltenname2, ...]	Wählt die Spalten einer oder mehrerer Tabellen, deren Inhalte in die Liste aufgenommen werden sollen; alle Spalten (*) oder die namentlich aufgeführten
FROM	Name der Tabelle oder Namen der Tabellen, aus denen die Daten der Ausgabe stammen sollen
SELECT ... **FROM** ... **(SELECT** ... **FROM** ... **WHERE** ...**) AS** tbl **WHERE** ...	Unterabfrage (subquery), die in eine äußere Abfrage eingebettet ist. Das Ergebnis der Unterabfrage wird wie eine Tabelle – hier mit Namen "tbl" – behandelt.
SELECT **DISTINCT**	Eliminiert Redundanzen, die in einer Tabellen auftreten können, Werte werden jeweils nur einmal angezeigt.
JOIN / INNER JOIN	Liefert nur die Datensätze zweier Tabellen, die gleiche Datenwerte enthalten
LEFT JOIN / LEFT OUTER JOIN	Liefert von der erstgenannten (linken) Tabelle alle Datensätze und von der zweiten Tabelle jene, deren Datenwerte mit denen der ersten Tabelle übereinstimmen
RIGHT JOIN / RIGHT OUTER JOIN	Liefert von der zweiten (rechten) Tabelle alle Datensätze und von der ersten Tabelle jene, deren Datenwerte mit denen der zweiten Tabelle übereinstimmen
WHERE	Bedingung, nach der Datensätze ausgewählt werden sollen
WHERE EXISTS (subquery **)** **WHERE NOT EXISTS (** subquery **)**	Die Bedingungen EXISTS prüft, ob die Suchbedingung einer Unterabfrage mindestens eine Zeile zurückliefert. NOT EXIST negiert die Bedingung.
WHERE ... IN (subquery **)** **WHERE NOT... IN (** subquery **)**	Der Wert des Datenfelds ist in der auswählten Menge vorhanden. Der Wert des Datenfelds ist in der auswählten Menge nicht vorhanden.
GROUP BY Spaltenname1 [,Spaltenname2, ...]	Gruppierung (Aggregation) nach Inhalt des genannten Feldes
ORDER BY Spaltenname1 [,Spaltenname2, ...] **ASC | DESC**	Sortierung nach Inhalt des genannten Feldes oder der genannten Felder ASC: aufsteigend; DESC: absteigend

Fortsetzung SQL-Syntax →

Syntax	Beschreibung
Datenmanipulation	
DELETE FROM Tabellenname	Löschen von Datensätzen in der genannten Tabelle
UPDATE Tabellenname **SET**	Aktualisiert Daten in Feldern einer Tabelle
INSERT INTO Tabellenname[(spalte1, spalte2, ...)] **VALUES** (Wert für Spalte 1 [, Wert für Spalte 2, ...]) oder **SELECT** ... **FROM** ... **WHERE**	Fügt Datensätze in die genannte Tabelle, die entweder mit festen Werten belegt oder Ergebnis eines SELECT-Befehls sind
Berechtigungen kontrollieren	
CREATE Benutzer | Rolle **IDENTIFIED BY** 'Passwort'	Erzeugt einen neuen Benutzer oder eine neue Rolle mit einem Passwort
GRANT Recht | Rolle **ON** *.* | Datenbank.* | Datenbank.Objekt **TO** Benutzer | Rolle [WITH GRANT OPTION]	Weist einem Benutzer oder einer Rolle ein Recht auf ein bestimmtes Datenbank-Objekt zu Weist einem Benutzer eine Rolle zu
REVOKE Rechte | Rollen **ON** *.* | Datenbank.* | Datenbank.Objekt **FROM** Benutzer | Rolle	Entzieht einem Benutzer oder einer Rolle ein Recht auf ein bestimmtes Datenbank-Objekt Entzieht einem Benutzer eine Rolle
Aggregatfunktionen	
AVG(Spaltenname)	Ermittelt das arithmetische Mittel aller Werte im angegebenen Feld
COUNT(Spaltenname | *)	Ermittelt die Anzahl der Datensätze mit Nicht-NULL-Werten im angegebenen Feld oder alle Datensätze der Tabelle (dann mit Operator *)
SUM(Spaltenname | Formel)	Ermittelt die Summe aller Werte im angegebenen Feld oder der Formelergebnisse
MIN(Spaltenname | Formel)	Ermittelt den kleinsten aller Werte im angegebenen Feld
MAX (Spaltenname | Formel)	Ermittelt den größten aller Werte im angegebenen Feld
Funktionen	
LEFT(Zeichenkette, Anzahlzeichen)	Liefert *Anzahlzeichen* der Zeichenkette von links.
RIGHT(Zeichenkette, Anzahlzeichen)	Liefert *Anzahlzeichen* der Zeichenkette von rechts.
CURRENT	Liefert das aktuelle Datum mit der aktuellen Uhrzeit
CONVERT(**time,**[DatumZeit])	Liefert die Uhrzeit aus einer DatumZeit-Angabe
DATE(Wert)	Wandelt einen Wert in ein Datum um
DAY(Datum)	Liefert den Tag des Monats aus dem angegebenen Datum
MONTH(Datum)	Liefert den Monat aus dem angegebenen Datum
TODAY	Liefert das aktuelle Datum
WEEKDAY(Datum)	Liefert den Tag der Woche aus dem angegebenen Datum
YEAR(Datum)	Liefert das Jahr aus dem angegebenen Datum
DATEADD(Datumsteil, Intervall, Datum)	Fügt einem Datum ein Intervall (ausgedrückt in den unter Datumsteil angegebenen Einheiten) hinzu
DATEDIFF(Datumsteil, Anfangsdatum, Enddatum) Datumsteile: **DAY, MONTH, YEAR**	Liefert Enddatum-Startdatum (ausgedrückt in den unter Datumsteil angegebenen Einheiten)
Operatoren	
AND	Logisches UND
LIKE	Überprüfung von Text auf Gleichheit wenn Platzhalter ("regular expressions") eingesetzt werden.
NOT	Logische Negation
OR	Logisches ODER
IS NULL	Überprüfung auf NULL
=	Test auf Gleichheit
>, >=, <, <=, < >	Test auf Ungleichheit
*	Multiplikation
/	Division
+	Addition, positives Vorzeichen
-	Subtraktion, negatives Vorzeichen

Stand 2021-09-30

Abschlussprüfung Sommer 2022
Lösungshinweise

Fachinformatiker/Fachinformatikerin Anwendungsentwicklung (AO 2020) 1201

1 Planen eines Softwareproduktes

Teil 2 der Abschlussprüfung

Allgemeine Korrekturhinweise

Die Lösungs- und Bewertungshinweise zu den einzelnen Handlungsschritten sind als Korrekturhilfen zu verstehen und erheben nicht in jedem Fall Anspruch auf Vollständigkeit und Ausschließlichkeit. Neben hier beispielhaft angeführten Lösungsmöglichkeiten sind auch andere sach- und fachgerechte Lösungsalternativen bzw. Darstellungsformen mit der vorgesehenen Punktzahl zu bewerten. Der Bewertungsspielraum des Korrektors (z. B. hinsichtlich der Berücksichtigung regionaler oder branchenspezifischer Gegebenheiten) bleibt unberührt.

Zu beachten ist die unterschiedliche Dimension der Aufgabenstellung (nennen – erklären – beschreiben – erläutern usw.).

Für die Bewertung gilt folgender Punkte-Noten-Schlüssel:

Note 1	=	100 – 92 Punkte	Note 2	= unter	92 – 81 Punkte
Note 3	= unter	81 – 67 Punkte	Note 4	= unter	67 – 50 Punkte
Note 5	= unter	50 – 30 Punkte	Note 6	= unter	30 – 0 Punkte

Die Vervielfältigung, Verbreitung und öffentliche Wiedergabe der Prüfungsaufgaben und Lösungen ist nicht gestattet. Zuwiderhandlungen werden zivil- und strafrechtlich (§§ 97 ff., 106 ff. UrhG) verfolgt. – © ZPA Nord-West 2022 – Alle Rechte vorbehalten!

1. Aufgabe (20 Punkte)

a) 8 Punkte

Erwartet werden Erläuterungen wie zum Beispiel:

Agil: Agiles Projektmanagement bezeichnet Vorgehensweisen, bei denen hohe Mitwirkung des Auftraggebers bei der Erstellung des Werks erforderlich ist.

Beispiele: Scrum, Kanban, XP, Prince2

Klassisch: Klassisches Projektmanagement bezeichnet Projektmanagementsysteme, die ein hohes Maß an Standardisierung aufweisen und als Referenz anerkannt sind.

Beispiele: Wasserfall-, Spiral- und V-Modell

Der Erfolg des Projektes hängt maßgeblich von der Akzeptanz der eigenen Mitarbeiter ab. Dazu ist von Ihnen ein innerbetriebliches Marketingkonzept zu erstellen.

b) 12 Punkte

Erwartet werden Beschreibungen zu Befürchtungen und entsprechenden Marketingmaßnahmen, wie z. B.:

Betroffener	Befürchtungen	Maßnahmen
Vorstand des Energieversorgers	– Keine Akzeptanz in der Belegschaft – Kostendimension schwer absehbar – Imageverlust bei Misserfolg der technischen Lösung	– Deutlich den Vorteil für ALLE kommunizieren – Ständiges Kosten-Controlling – Ständiges Technik-Controlling
Stromableser	– Neue technische Herausforderungen – Arbeitsverdichtung – Die Einführung führt zum Verlust des Arbeitsplatzes	– Schulungen anbieten – Gehaltsanpassung sowie optimieren weiterer sozialer Benefits – Aussprechen einer Beschäftigungsgarantie

Andere richtige Befürchtungen und geeignete Gegenmaßnahmen sind ebenfalls zu bewerten.

2. Aufgabe (25 Punkte)

a) 4 Punkte

Erwartet werden Nennungen, wie z. B.:

Vorteile:
- Bessere Kostenkontrolle (Festpreis)
- Günstiger durch Wettbewerb
- Geringe Bindung interner Ressourcen

Nachteile:
- Möglicher Verlust sicherheitsrelevanter Informationen
- Schwieriger Transfer von internen Abläufen
- Abhängigkeit von Fremdunternehmen

b) 12 Punkte

Mögliche funktionale Anforderungen können z. B. sein:
- Lauffähigkeit auf gängigen mobilen Systemen (iOS, Android etc.)
- Authentifizierung gewährleisten
- Push-Benachrichtigung, z. B. für Erinnerungen, Termine, bei auftretenden Fehlern
- Schnittstellen-Definition
- Validierung der Eingaben

Mögliche nichtfunktionale Anforderungen können z. B. sein:
- Verfügbarkeit
- Zuverlässigkeit
- Usability
- Fehlertoleranz
- Sicherstellung der Datenintegrität

c) 9 Punkte
- Durch flexible Datenmodelle sind neue Funktionen schneller und einfacher umsetzbar.
- Bei komplexeren Datenstrukturen können Abfragen schneller durchgeführt werden.
- Einfachere Entwicklung durch zum Programm passende Datenstrukturen.

Weitere Vorteile, die sich aus dem Text ergeben, sind ebenfalls möglich

3. Aufgabe (30 Punkte)

a) 7 Punkte

SmartMeter	0,5 Punkte
– meterNumber : String – counter : Double oder Integer	1 Punkt 1 Punkt
+ SmartMeter(meterNumber : String, counter : Double) + setCounter(counter : Double) : void + getCounter() : Double	1,5 Punkte 1,5 Punkte 1,5 Punkte

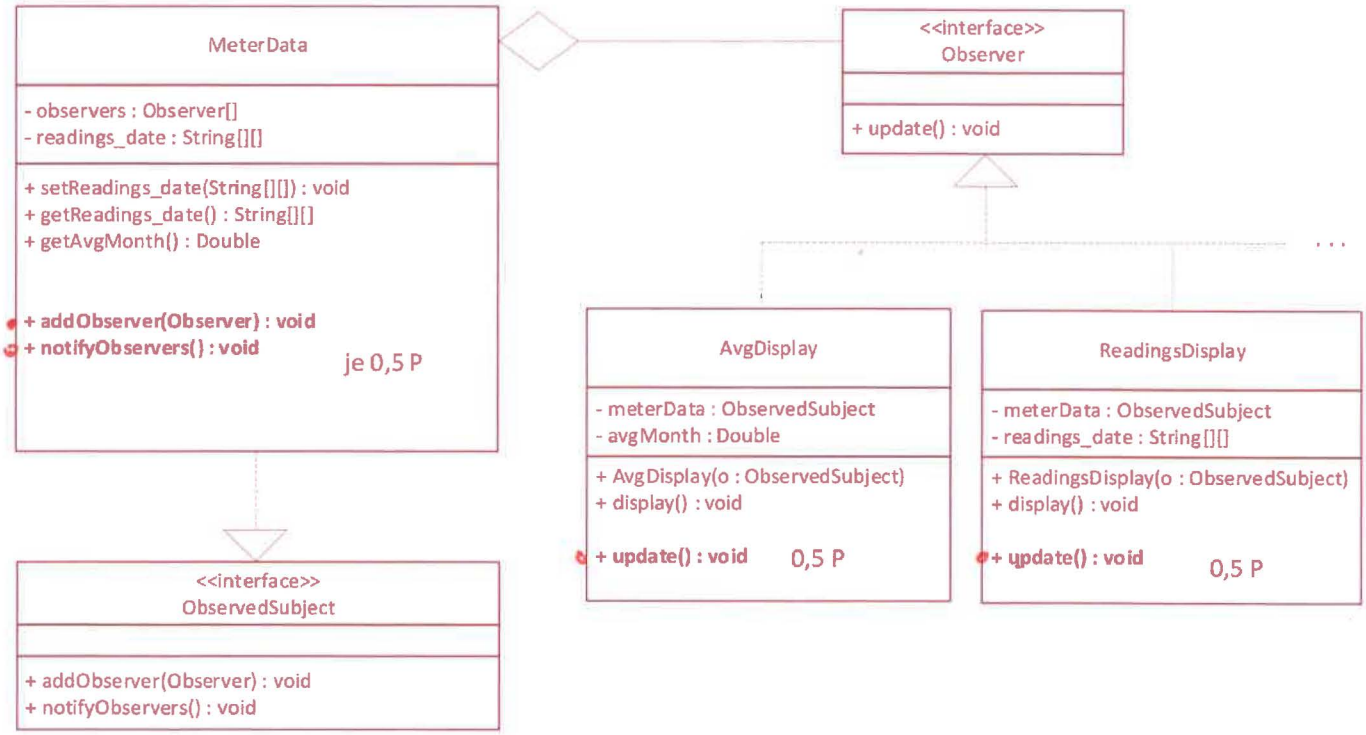
ba) 4 Punkte

1 Punkt Symbol:

Aggregation: „has a-Beziehung", die Klasse MeterData kann mehrere Observerobjekte haben (z. B. die Anzeigen oben oder andere graphische Auswertungsanzeigen). 3 Punkte

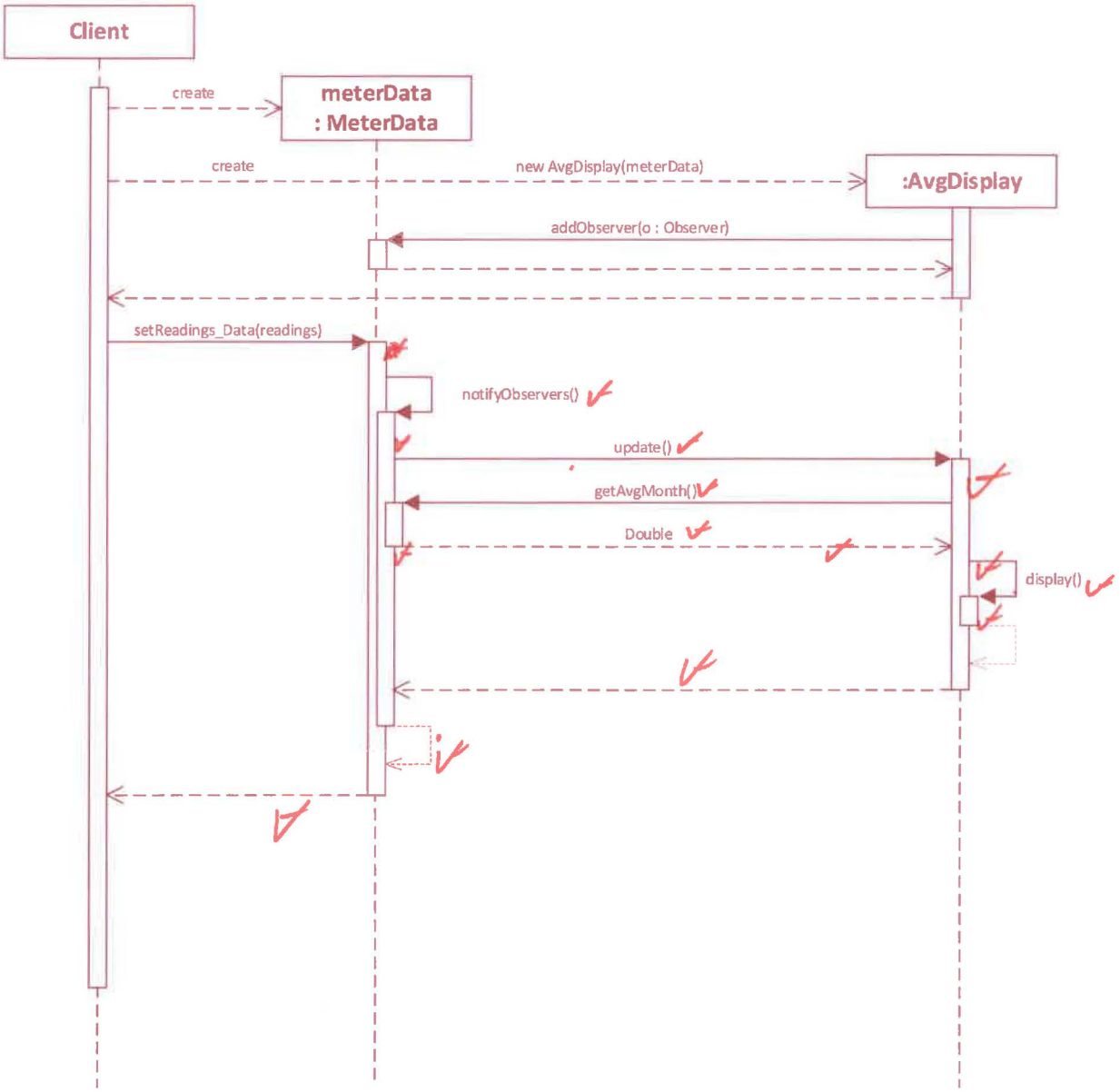
Hinweis: Multiplizitäten müssen nicht angegeben werden

bb) 2 Punkte
je 0,5 Punkte pro implementierte Methode



bc) 4 Punkte
Realisierung/Schnittstellenimplementierung:
Die Klasse AvgDisplay muss die abstrakte Methode update des implementierten Interface Observers überschreiben.
Damit bekommt AvgDisplay die Rolle eines Observers zugewiesen. Die Methode notifyObservers der Klasse MeterData ruft auf allen Objekten mit Observer-Rolle update auf. Dank Polymorphie ist jede Implementierung von update spezifisch für das jeweilige Objekt.

ca) 9 Punkte
9 Nachrichten+Antworten, 4 Balken, 5 Beschriftungen je 0,5 Punkte



cb) 4 Punkte
1 Punkt Methodenkopf, 2 Punkte Schleife, 1 Punkt Methodenaufruf

```
public void notifyObservers() {
    for (Observer o: observers) {
        o.update();
    }
}
```

4. Aufgabe (25 Punkte)

a) 12 Punkte

Zeitersparnis: Ob Dienstleister oder Kunde, beide profitieren von strukturierten Prozessen, die mithilfe des Ticketsystems abgebildet werden. Zudem lassen sich Mitarbeiter gezielt schulen, was den Aufwand für Nachschulungen reduziert. Dank der Echtzeit-Auswertung können schnell unnötige Schritte eliminiert und Arbeitsabläufe optimiert werden.

Echtzeit-Übersicht: Alle aktuellen und vergangenen Tickets und deren Status werden übersichtlich angeordnet. Zudem gibt es Reportings und statistische Erhebungen wie Ticketaufkommen, Reaktionszeiten oder Aussagen zur Bearbeitungsqualität. Damit wird das Ticketsystem zur Wissensdatenbank mit Suchfunktion.

Kundenfreundlichkeit: Je nach Service Level Agreement bearbeitet der Support die Anfragen. Durch diese Festlegung wird dem Kunden eine bestimmte Response-Zeit zugesichert. Das schafft Vertrauen und auch Nähe zum Kunden. Durch die transparente Abrechnung und detaillierten Nachverfolgung aller Tickets steigt auch das Verständnis für die erbrachte Leistung.

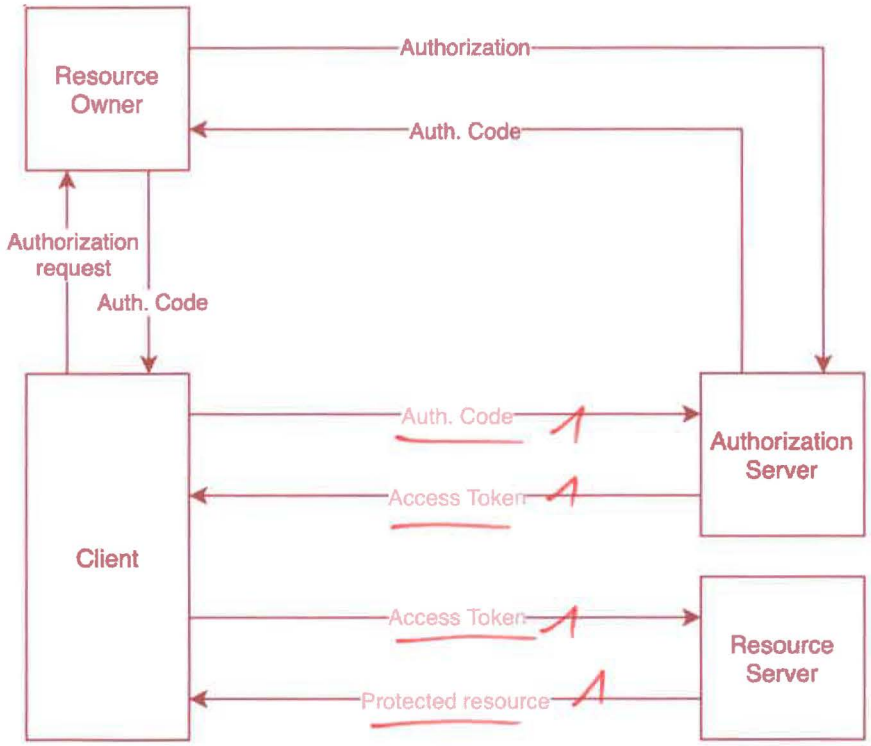
Simple Nutzung: Ticketsysteme sind 24 Stunden am Tag verfügbar. Sie lassen sich über Mail oder ein eigenes Portal abrufen. Durch die strukturierte Aufbereitung, die automatische Ticket-Zuteilung und die automatische Benachrichtigung bei Statusänderungen behält jeder Nutzer den Überblick. Das erleichtert die Zusammenarbeit.

Schnittstellen: Viele Systemen bieten zahlreiche Schnittstellen zu Fremdprogrammen wie dem ERP- oder CRM-System. Aber auch Collaborationtools wie Videokonferenzen und klassische Telefonanlagen lassen sich integrieren. Das trägt ebenfalls zu einem hohen Kundenservice bei, da alle Systeme miteinander kommunizieren können.

Nachteile von Ticketsystemen

Neben den genannten Vorteilen gibt es aber auch einige Haken. Einer zeigt sich beispielsweise in den Kosten, die mit der Einführung des Ticketsystems verbunden sind. Um das System dauerhaft nutzen zu können, braucht es zudem kompetentes Personal. Die Experten müssen Änderungen vornehmen und die Lösung warten. Damit verbunden sind oft auch leistungsstärkere Server. Ein weiterer Aspekt zeigt sich in der Transparenz: Durch die genaue Buchung der Arbeitszeit pro Ticket kann der Vorwurf der Überwachung aufkommen. Mitarbeiter, die das als Verhaltens- oder Leistungskontrolle wahrnehmen, können schnell sehr unzufrieden sein und im schlimmsten Fall kündigen. Das kostet Zeit (Recruitingprozess), Ressourcen (Einarbeitung) und Geld (Schulungen). Auch ist zu prüfen, ob Ticketsysteme im Widerspruch zum Datenschutz oder Arbeitsrecht stehen.

b) 4 Punkte



c) 9 Punkte

Durch den Einsatz eines Zertifikates werden die folgenden Sicherheitsmechanismen erreicht:

- Der Einsatz von Zertifikaten ist relativ einfach und kostengünstig.
- Sender und Empfänger sind authentifiziert.
- Die Übertragung wird kryptografisch geschützt übermittelt (Verschlüsselung).
- Die Integrität der Daten wird sichergestellt.
- Die Zugangskontrolle ist gesichert.